

DATA COMMUNICATIONS INTERFACES

The present invention relates to data communications interfaces for nodes of a data processing network.

5

A conventional data processing network comprises a plurality of host computer systems and a plurality of attached devices all interconnected by an intervening network architecture such as an Ethernet architecture. The network architecture typically comprises one or more data communications switches. The host computer systems and the attached devices each form a

10 node in the data processing network. Each host computer system typically comprises a plurality of central processing units and data storage memory device interconnected by a bus architecture such as a PCI bus architecture. A network adapter is also connected to the bus architecture for communicating data between the host computer system and other nodes in the data processing network via the network architecture. It would be desirable for the network
15 adapter to facilitate transfer of data between the bus architecture of the computer system and the network architecture as efficiently as possible.

In accordance with the present invention, there is now provided a data communications interface for a node of a data processing network, the interface comprising: a transmission
20 channel for communicating data from the node to the network; a transmission processor connected to the transmission channel for controlling flow of data through the transmission channel; a reception channel for communicating data from the network to the node; a reception processor connected to the reception channel for controlling flow of data through the reception channel; a shared memory; and, a local bus providing access to the shared
25 memory by the transmission and reception processors.

This arrangement advantageously permits the transmission and reception processors to access the shared memory independently of the transmission and reception channels, thereby maximizing processing efficiency in the transmission and reception processors and hence
30 efficiency in transfer of data between the node and the data network.

The shared memory preferably comprises a store for control information to be used by the transmission and reception processors in controlling said flows of data.

Preferred embodiments of the present invention, comprise a communication path for communicating information between the transmission and reception processors via the shared memory.

5

Particularly preferred embodiments of the present invention comprise bus interface logic for connecting the local bus, the transmission data channel, and the reception data channel to a bus architecture of the node.

- 10 Especially preferred embodiments of the present invention comprise: a transmission control path; transmission segmentation logic for receiving a data frame, comprising a transmission header and a transmission payload, from the node and supplying the transmission payload to the transmission channel and the transmission header to the transmission control path; the transmission processor being located in the transmission control path for controlling
- 15 communication of data from the transmission payload to the network via the transmission channel in dependence on the transmission header; a reception control path; reception segmentation logic for receiving a data packet, comprising a reception header and a reception payload, from the network and supplying the reception payload to the reception channel and the reception header to the reception control path; and, the reception processor being located
- 20 in the reception control path for controlling communication of data from the reception payload to the node via the reception channel in dependence on the reception header.

The present invention extends to a network interface card for insertion into a computer system comprising a printed circuit board and a data communications interface as

- 25 hereinbefore described mounted on the printed circuit board. The present invention also extends to an application specific integrated circuit comprising a data communications interface as hereinbefore described. The present invention further extends to a computer system comprising: a central processing unit; a memory; a data communications interface as hereinbefore described; and, a bus architecture interconnecting the central processing unit, the
- 30 memory, and the data communications interface. Furthermore, the present invention extends to a data processing network comprising a plurality of such computer systems and a network architecture interconnecting the computer systems.

Viewing the present invention from another aspect, there is now provided a method for communicating data to and from a node of a data processing network, the method comprising: communicating data from the node to the network via a transmission channel; controlling flow of data through the transmission channel via a transmission processor
5 connected to the transmission channel; communicating data from the network to the node via a reception channel; controlling flow of data through the reception channel via a reception processor connected to the reception channel; and, providing access to a shared memory by the transmission and reception processors via a local bus.

10 Such a method preferably comprises storing, in the shared memory, control information to be used by the transmission and reception processors in controlling said flows of data. Such a method may also or alternatively comprise communicating, via the shared memory, information between the transmission and reception processors.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199

Figure 8 is a block diagram of a logical receive path through the ISOC.

Referring first to Figure 1, an example of a data processing network embodying the present invention comprises a plurality of host computer systems 10 and a plurality of attached
5 devices 20 interconnected by an intervening network architecture 30 such as an Ethernet or an InfiniBand network architecture (InfiniBand is a trade mark of the InfiniBand Trade Association). The network architecture 30 typically comprises a plurality of data communications switches 40. The host computer systems 10 and the attached devices 20 each form a node in the data processing network. Each host computer system 10 comprises a
10 plurality of central processing units (CPUs) 50, and a memory 60 interconnected by a bus architecture 70 such as a PCI bus architecture. A network adapter 80 is also connected to the bus architecture for communicating data between the host computer system 10 and other nodes in the data processing network via the network architecture 30.

15 Referring now to Figure 2, in particularly preferred embodiments of the present invention, the network adapter 80 comprises a pluggable option card having a connector such as an edge connector for removable insertion into the bus architecture 70 of the host computer system 10. The option card carries an Application Specific Integrated Circuit (ASIC) or Integrated System on a Chip (ISOC) 120 connectable to the bus architecture 70 via the connector 170,
20 one or more third level memory modules 250 connected to the ISOC 120, and an interposer 260 connected to the ISOC 120 for communicating data between the media of the network architecture 30 and the ISOC 120. The interposer 260 provides a physical connection to the network 30. In some embodiments of the present invention, the interposer 260 may be implemented in a single ASIC. However, in other embodiments of the present invention, the
25 interposer 260 may be implemented by multiple components. For example, if the network 30 comprises an optical network, the interposer 260 may comprise a retimer driving a separate optical transceiver. The memory 250 may be implemented by SRAM, SDRAM, or a combination thereof. Other forms of memory may also be employed in the implementation of memory 250. The ISOC includes first and second memory. The memory subsystem of the
30 adapter 80 will be described shortly. As will become apparent from the following description, this arrangement provides: improved performance of distributed applications operating on the data processing network; improved system scalability; compatibility with a range of communication protocols; and reduced processing requirements in the host computer system.

More specifically, this arrangement permits coexistence of heterogeneous communication protocols between the adapters 80 and the host systems 10. Such protocols can serve various applications, use the same adapter 80, and use a predefined set of data structures thereby enhancing data transfers between the host and the adapter 80. The number of application
5 channels that can be opened in parallel is determined by the amount of memory resources allocated to the adapter 80 and is independent of processing power embedded in the adapter. It will be appreciated from the following that the ISOC 120 concept of integrating multiple components into a single integrated circuit chip component advantageously minimizes manufacturing costs and in provides reusable system building blocks. However, it will also be
10 appreciated that in other embodiments of the present invention, the elements of the ISOC 120 may be implemented by discrete components.

In the following description, the term Frame refers to data units or messages transferred between software running on the host computer system 10 and the adapter 80. Each Frame
15 comprises a Frame Header and a data payload. The data payload may contain user data, high level protocol header data, acknowledgments, flow control or any combination thereof. The contents of the Frame Header will be described in detail shortly. The adapter 80 processes only the Frame Header. The adapter 80 may fragment Frames into smaller packets which are more efficiently transported on the network architecture 30. However, such fragmentation
20 generally does not transform the data payload.

In particularly preferred embodiment of the present invention, data is transported on the network architecture 30 in atomic units hereinafter referred to as Packets. Each Packet comprises route information followed by hardware header data and payload data. In a typical
25 example of the present invention, a packet size of up to 1024 bytes is employed. Frames of larger size are fragmented into 1024 byte packets. It will be appreciated that in other embodiments of the present invention, different packet sizes may be employed.

In a preferred embodiment of the present invention, communications between the adapter 80
30 and multiple applications running on the host computer system 10 are effected via a Logical Communication Port architecture (LCP). The adapter 80 comprises a memory hierarchy which allows optimization of access latency to different internal data structures. This memory hierarchy will be described shortly. In preferred embodiments of the present invention, the

adapter 80 provides separate paths for outbound (TX) data destined for the network architecture 30 and inbound (RX) data destined for the host computer system 10. Each path includes its own data transfer engine, header processing logic and network architecture interface. These paths will also be described in detail shortly.

5

LCP Architecture

Referring now to Figure 3, the LCP architecture defines a framework for the interface between local consumers running on the host computer system 10 and the adapter 80.

10 Examples of such consumers include both applications and threads. The computer system 10 can be subdivided into a user application space 90 and a kernel space 110. The LCP architecture provides each consumer with a logical port into the network architecture 30. This port can be accessed directly from a user space 90. In particularly preferred embodiments of the present invention, a hardware protection mechanism takes care of access permission. An
15 LCP registration is performed by in the kernel space 110 prior to transfer of data frames. The LCP architecture need not define a communication protocol. Rather, it defines an interface between the applications and the adapter 80 for transfer of data and control information. Communication protocol details may be instead set by the application and program code executing in the adapter 80. The number of channels that can be used on the adapter 80 is
20 limited only by the amount of memory on the adapter card 80 available for LCP related information. Each LCP port can be programmable to have a specific set of features. The set of features is selected according to the specific protocol to best support data transfer between the memory 60 in the host computer system and the adapter 80. Various communication protocols can be supported simultaneously, with each protocol using a different LCP port.

25

The LCP architecture comprises LCP Clients 100, an LCP Manager 130 resident in the kernel space 130, and one or more LCP Contexts 140 resident in the adapter 80.

Each LCP Client 100 is a unidirectional application end point connected to an LCP port. An
30 LCP client 100 can be located in the user application space 90 or in the kernel 110. In operation, each LCP client 100 produces commands and data to be read from the memory 60 and transferred by the adapter 80 via a TX LCP channel, or consumes data transferred by the adapter 80 to the memory 60 via an RX LCP channel.

The LCP Manager 130 is a trusted component that services request for LCP channel allocations and deallocations and for registration of read/write areas in the memory 60 for each channel. The LCP Manager 130 allows a user space application to use resources of the adapter 80 without compromising other communication operations, applications, or the operating system of the host computer system 10.

Each LCP Context 140 is the set of control information required by the adapter 80 to service a specific LCP Client 100. The LCP Context 140 may include LCP channel attributes which are constant throughout existence of the channel, such as possible commands, pointer structure, and buffer descriptor definitions. The LCP Context 140 may also include specific LCP service information for the LCP channel, such as the amount of data waiting for service, and the next address to access for the related LCP channel. The LCP context 140 is stored in memory resident in the adapter 80 to enable fast LCP context switching when the adapter 80 stops servicing one channel and starts servicing another channel.

An LCP Client 100 requiring initiation of an LCP port turns to the LCP Manager 130 and requests the allocation of an LCP channel. The LCP channel attributes are determined at this time and prescribe the behavior of the LCP port and the operations that the LCP Client 100 is authorized to perform in association with the LCP port. The LCP Client 100 is granted an address that will be used to access the adapter 80 in a unique and secure way. This address is known as a Doorbell Address.

The LCP Manager 130 is also responsible for registering areas of the host memory 60 to enable virtual to physical address translation by the adapter, and to allow user space clients to access these host memory areas without tampering with other programs.

Registration of new buffers and deregistration of previous buffers can be requested by each LCP Client 100 during run-time. Such a change, requires a sequence of information exchanges between the LCP Client 100, the LCP Manager 130, and the adapter 80.

Each LCP Client 100 and port are associated with an LCP Context 140 that provides all the information required by the adapter 80 to service pending requests sent by the LCP port for command execution.

- 5 To initiate memory transfers between the LCP Client 100 and the adapter 80, and initiate transmission of frames, the LCP Client 100 prepares descriptors holding the information for a specific operation. The LCP Client 100 then performs an I/O write to the Doorbell address mapped to the adapter 80. Writing to the Doorbell address updates the LCP Context 140 on the adapter 80, adding the new request for execution.

10

The adapter 80 arbitrates between various transmit LCP ports that have pending requests, and selects the next one to be serviced.

- 15 On receipt of data, the Frame and LCP for a received packet are identified. Descriptors are generated to define the operation required for the receive LCP. Execution of these descriptors by an LCP Engine of the adapter 80 stores the incoming data in an appropriate data buffer allocated to the LCP channel in the memory 60 of the host computer system 10.

- 20 For each LCP channel serviced, the adapter 80 loads the associated LCP context information and uses this information to perform the desired set of data transfers. The adapter 80 then continues on to process the next selected LCP Context 140.

Adapter Memory

- 25 Referring now to Figure 3, and as mentioned earlier, the ISOC 120 comprises a first memory space 220 and 230 and a second memory space 240 and the adapter 80 further comprises a third level memory. The first, second, and third memory spaces for part of a memory subsystem 210 of the adapter 80. In a preferred embodiment of the present invention, the ISOC 120 comprises a TX processor (TX MPC) 150 dedicated to data transmission
- 30 operations and an RX processor (RX MPC) 160 dedicated to data reception operation. In particularly preferred embodiments of the present invention, processors 150 and 160 are implemented by Reduced Instruction Set Computing (RISC) microprocessors such as IBM PowerPC 405 RISC microprocessors. Within the memory subsystem 210, the ISOC 120

comprises, in addition to the first and second memory spaces, a data cache 180 and an instruction cache 170 associated with TX processor 150, together with a second data cache 190 and second instruction cache 190 associated with RX processor 160. The difference between the three levels is the size of memory and the associated access time. As will become
5 apparent shortly, the memory subsystem 210 facilitates: convenient access to instruction and data by both the TX processor 150 and the RX processor 160; scalability; and sharing of resources between the TX processor 150 and the RX processor 160 in the interests of reducing manufacturing costs.

10 The first level memory spaces (M1) 220 and 230 comprise a TX-M1 memory space 220 and RX-M1 memory space 230. The TX-M1 memory 220 can be accessed only by the TX processor 150 and the RX-M1 memory 230 can be accessed only by the RX processor 160. In operation the first level memory spaces 220 and 230 are used to hold temporary data structures, header templates, stacks, etc. The first level memory spaces 220 and 230 both
15 react to zero wait states. Each one of the first level memory spaces 220 and 230 is connected only to the data interface of the corresponding one of the processors 150 and 160 and not to the instruction interface. This arrangement enables both cacheable and non-cacheable first level memory areas available while maintaining efficient access to data in the first level memory spaces 230 and 240.

20 The second level memory space (M2) 240 is a shared memory available to both processors 150 and 160, other components of the adapter 80, and to the host computer system 10. Access to the second level memory space 240 is slower than access to the first level memory areas 220 and 230 because the second level memory space 240 is used by more agent via a shared
25 internal bus. The third level memory space 250 is also a shared resource. In particularly preferred embodiments of the present invention the adapter 80 comprises a computer peripheral circuit card on which the first level memory spaces 220 and 230 and the second level memory space 240 are both integrated on the same ASIC as the processors 150 and 160.

30 The shared memory spaces 240 and 250 are generally used for data types that do not require fast and frequent access cycles. Such data types include LCP contexts 140 and virtual address

translation tables. The shared memory spaces 240 and 250 are accessible to both instruction and data interfaces of the processors 150 and 160.

The adapter 80 handles transmission and reception data flows separately. The separate

5 processor 150 and 160 for the transmission and reception path avoids the overhead of switching between task, isolates temporary processing loads in one path from the other path, and facilitates use of two embedded processors to process incoming and outgoing data streams. Referring now to Figure 5, the ISOC 120 comprises transmission path logic 280 and reception path logic 290, and shared logic 300. The transmission path logic 280 comprises an
10 LCP TX engine 310 for decoding specifics of each LCP channel and fetching LCP related commands for execution; TX logic 320 for controlling transfer of frames into the adapter 80 , the aforementioned TX processor 150 for managing TX frame and packet processing; the aforementioned first level TX memory 220 for holding instructions and temporary data structures; and link logic 330; and logic for assisting the TX processor 150 in managing the
15 data flow and packet processing such as routing processing for fragmentation of frames into data packets. The TX processor 150 processes tasks in series based on a polling only scheme in which the processor is interrupted only on exceptions and errors. The first level TX memory 220 is employed by the processor 150 for communicating with TX logic 320. The reception path logic 290 comprises link logic 340; hardware for assisting the aforementioned
20 RX processor 160 in processing headers of incoming packets and transformation or assembly of such packets into frames; the aforementioned RX processor 160 for RX frame and packet processing; the aforementioned first level RX memory 230 for holding instructions; RX logic 350 for controlling transfer of frames from the network architecture 30; and an LCP RX engine 360 for decoding the specifics of each LCP channel, storing the incoming data in the
25 related LCP data structures in the memory 60 of the host computer system, and accepting and registering pointers to empty frame buffers as they are provided by the LCP Client 100 for use by the adapter 80. The RX processor 160 processes tasks in series using a polling only scheme in which the RX processor 160 is interrupted only on exceptions or errors. The level
30 1 RX memory 230 is used by the RX processor 160 to communicate with the RX logic 350.

As mentioned earlier, the ISOC approach permits reduction in manufacturing costs associated with the adapter 80 and the other components thereof, such as the circuit board and the other supporting modules. The ISOC approach also increases simplicity of the adapter 80, thereby

increasing reliability. The number of connections between elements of the ISOC 120 is effectively unlimited. Therefore, multiple and wide interconnect paths can be implemented. In the interests of reducing data processing overheads in the host computer system 10, data transfer operations to and from the host memory 60 are predominantly performed by the

5 ISOC 120. The ISOC 120 also performs processing of the header of incoming and outgoing packets. During transmission, the ISOC 120 builds the header and routes it to the network architecture 30. During reception, the adapter 80 processes the header in order to determine its location in the system's memory. The level 1 memories 220 and 230 are zero wait state memories providing processor data space such as stack, templates, tables, and temporary

10 storage locations. In especially preferred embodiments of the present invention, the transmission path logic 280, reception path logic 290, and shared logic 300 are built from smaller logic elements referred to as cores. The term core is used because there elements are designed as individual pieces of logic which have stand-alone properties enabling them to be used for different applications.

15 As indicated earlier, the transmission path logic 280 is responsible for processing transmission or outgoing frames. Frame transmission is initiated via the bus architecture 70 by a CPU such as CPU 50 of the host computer system 10. The ISOC 120 comprises bus interface logic 370 for communicating with the bus architecture 70. The ISOC 120 also

20 comprises bus bridging logic 390 connecting the bus interface logic 370 to a processor local bus (PLB) 390 of the ISOC 120. The TX LCP engine 310 fetches commands and frames from the host memory 60. The TX processor 150 processes the header of each frame into a format suitable for transmission as packets on the network architecture 30. The TX logic 320 transfer the frame data without modification. The link logic 330 processes each packet to be

25 transmitted into a final form for transmission on the network architecture 30. The link logic 330 may comprises one or more ports each connectable to the network architecture 30.

As indicated earlier, the reception path logic 290 is responsible for processing incoming packets. Initially, packets received from the network architecture 30 are processed by link

30 logic 340. Link logic 340 recreates the packet in a header and payload format. To determine the packet format and its destination in the host memory 60, the header is processing by the RX processor 230. The link logic 340 may comprises one or more ports each connectable to

the network architecture 30. The RX LCP engine is responsible for transferring the data into the host memory 60 via the bus architecture 70.

The transmission path logic 280 comprises a HeaderIn first in- first out memory (FIFO) 400 between the TX LCP engine 310 and the TX processor 220. The reception path logic comprises a HeaderOut FIFO 410 between the RX processor 230 and the RX LCP engine 360. Additional FIFOs and queues are provided in the TX logic 320 and the RX logic 350. These FIFOs and queues will be described shortly.

The shared logic 300 comprises all logical elements shared by the transmission path logic 280 and the reception path logic 290. These elements include the aforementioned bus interface logic 370, bus bridging logic 380, PLB 390, second level memory 240 and a controller 420 for providing access to the remote third level memory 250. The bus interface logic 370 operates as both master and slave on the bus architecture 70. As a slave, the bus interface logic allows the CPU 50 to access the second level memory 240, the third level memory 250 via the controller 420, and also configuration registers and status registers of the ISOC 120. Such registers can generally be accessed by the CPU 50, the TX processor 150 and the RX processor 160. As a master, the bus interface logic allows the TX LCP engine 310 and the RX LCP engine 360 to access the memory 60 of the host computer system 10. In Figure 5, "M" denotes a master connection and "S" denotes a slave connection.

Packet Flow

Referring now to Figure 6, packet flow through the ISOC 120 is generally symmetrical. In other words, the general structure of flow is similar in both transmit and receive directions. The ISOC 120 can be regarded as comprising first interface logic 440; a first control logic 460; processor logic 480; second control logic 470; and second interface logic 450. Packets are processed in the following manner:

- A. In the transmit direction, information is brought into the ISOC 120 from the bus architecture 70 through the first interface logic. In the receive direction, information is brought into the ISOC 120 from the network architecture 30 through the second interface logic 450.

B. In the transmit direction, information brought into the ISOC 120 through the first interface logic 440 is processed by the first control logic 460. In the receive direction, information brought into the ISOC through the second interface logic 450 is processed by the second control logic 470.

C. In the transmit direction, a frame header is extracted for an outgoing frame at the first control logic 460 and processed by the processor logic 480. The processor logic 480 generates instructions for the second control logic 470 based on the frame header. The payload of the outgoing frame is passed to the second interface logic 470. In the receive direction, a frame header is extracted from an incoming frame at the second control logic 470 and processed by the processor logic 480. The processor logic 480 generates instructions for the first control logic 460 based on the frame header. The payload of the incoming frame is passed to the first control logic 460. In both directions, the processor 480 is not directly handling payload data.

D. In the transmit direction, the second control logic 470 packages the outgoing payload data according to the instructions received from the processor logic 480. In the receive direction, the first control logic 460 packages the incoming payload according to the instructions received from the processor logic 480.

E. In the transmit direction, the information is moved through the second interface logic 450 to its destination via the network architecture 30. In the receive direction, the information is moved through the first interface logic to its destination via the bus architecture 70.

An interface to software operating on the host computer system 10 is shown at 430. Similarly, interfaces to microcode operating on the processor inputs and outputs is shown at 490 and 500.

Transmit flow

Referring to Figure 7, what follows now is a more detailed description of one example of a flow of transmit data frames through the ISOC 120. The ISOC 120 can be divided into an LCP context domain 510, a frame domain 520 and a network domain 530 based on the various formats of information within the ISOC 120. The TX LCP engine 310 comprises an LCP requests FIFO 550, Direct Memory Access (DMA) logic 560, frame logic 580, and the

5 LCP requests FIFO 550, Direct Memory Access (DMA) logic 560, frame logic 580, and the aforementioned LCP context logic 140. The LCP request FIFO 550, DMA logic 560, and LCP TX Context logic 590 reside in the LCP context domain 510. The frame logic 580 resides in the frame domain 520. The TX logic 320, first level TX memory space 220, and TX processor 150 straddle the boundary between the frame domain 520 and the network

10 domain 530. The TX link logic 330 resides in the network domain 530. In particularly preferred embodiments of the present invention, the HeaderIn FIFO 400 is integral to the first level TX memory space 220. In general, an application executing on the host computer system 10 creates a frame. The frame is then transmitted using a TX LCP channel on the adapter 80. Handshaking between the application and the adapter 80 assumes a prior

15 initialization performed by the LCP Manager 130. To add an LCP Service Request, an LCP Client 100 informs the adapter 80 that one or more additional transmit frames are ready to be executed. This is performed by writing to a control word in to a Doorbell. The Doorbell's addresses are allocated in such a way that the write operation is translated into a physical write cycle on the bus architecture 70, using an address that is uniquely associated with the

20 LCP port and protected from access by other processes. The adapter 80 detects the write operation and logs the new request by incrementing an entry of previous requests for the specific LCP Client 100. This is part of the related LCP Context 140. An arbitration list, retained in the memory subsystem 210 of the adapter 80 is also updated. In a simple example, arbitration uses the aforementioned FIFO scheme 550 between all transmit LCP channels

25 having pending requests. While one LCP channel is serviced, the next LCP channel is selected. The service cycle begins when the corresponding LCP Context is loaded into the TX LCP engine 310. The LCP Context 140 is then accessed to derive atomic operations for servicing the LCP channel and to determine parameters for such operations. For example, such atomic operations may be based on LCP channel attributes recorded in the LCP Context

30 140. A complete service cycle typically includes a set of activities performed by the adapter 80 to fetch and execute a plurality of atomic descriptors created by the LCP Client 100. In the case of a TX LCP channel, the service cycle generally includes reading multiple frames from the host memory 60 into the memory subsystem 210 of the adapter 80. Upon conclusion, all

the LCP Context information requiring modification (in other words, the LCP Service Information) is updated in the memory subsystem 210 of the adapter 80. In general, the first action performed by the adapter 80 within the LCP Service cycle, is to fetch the next descriptor to be processed.

5

Processing of transmission frames by the ISOC 120 typically includes the following steps:

A. Fetching the subsequent LCP port frame descriptor.

10 The address of the next descriptor to be fetched is stored as parts of the LCP channel's Context 140. The adapter 80 reads the descriptor from host memory 60 and decodes the descriptor based on the LCP channel attributes. The descriptor defines the size of the new frame header, the size of the data payload, and the location of these items.

15 B. Conversion of virtual address to physical address.

If a data buffer is referenced by virtual memory addresses in an application, the address should go through an additional process of address translation. In this case, the virtual address used by the application is translated into a physical address usable by the adapter 80 while it access the host memory 60. This is done by monitoring page boundary crossings and using physical page location information written by the LCP manager 130 into the memory subsystem 210 of the adapter 80. The virtual to physical translation process serves also as a security measure in cases where a descriptor table is created by an LCP client 100 which is not trusted. This prevents
20 unauthorized access to unrelated areas of the host memory 60.

25

C. Reading the frame header.

Using physical addressing, the header and payload data of the TX frame are read from
30 buffers in the host memory 60. The header is then stored in the TX HeaderIn FIFO 400. When the header fetch is completed, the adapter 80 sets an internal flag indicating that processing of the header can be initiated by the TX processor 150.

D. Reading the frame data.

The payload data is read from the host memory 60 and stored by the adapter 80 in a data FIFO 570. The data FIFO 570 is shown in Figure 7 as resident in the TX logic 320. However, the data FIFO 570 may also be integral to the first level TX memory space 220. Data read transactions continue until all data to be transmitted is stored in the memory subsystem 210 of the adapter 80. Following completion of the read operation, a status indication is returned to the LCP Client 100. Note that processing of the header can start as soon as the header has been read into the HeaderIn FIFO 400. There is no need to wait for the whole data to be read.

E. Processing the frame header

The header processing is performed by the TX processor 150. Header processing is protocol dependent and involves protocol information external to the LCP architecture. The TX processor 150 runs TX protocol header microcode and accesses routing tables and other relevant information already stored in the memory subsystem 210 of the adapter 80 during a protocol and routing initialization sequence. When the TX processor 150 receives an indication that a new header is waiting in the HeaderIn FIFO 400, it starts the header processing. The header processing produces one or more packet headers which are in the format employed to send packets over the network architecture 30 and include routing information. If the payload size is larger than a maximum packet size allowed by the network architecture 30, the payload is fragmented by generating several packet headers each used in connection with consecutive data segments of the original payload data to form packets for communication over the network architecture 30.

F. Queuing the packet header for transmission

A command defining the number of header words and the number of data words for a packet and the packet header itself are written by the TX processor 150 to a TX HeaderOut FIFO 540 in the first level memory space 220.

G. Merging packet header and packet data for transmission.

Transmission of a packet on the network architecture 30 is triggered whenever a command is ready in the HeaderOut FIFO 540, and the data FIFO 570 contains
 5 enough data to complete the transmission of the related packet. A Cyclic Redundancy Check (CRC) may be added to the header and data of each packet. Each complete packet is transferred to the network architecture 30 via the TX link logic 330.

The transmission process for each frame is completed when all the frame data is transmitted
 10 on the network architecture 30, by means of one or more packets. For each frame processed by the adapter 80, a status may be returned to the application via a second LCP Client 100. This status indicates the completion of the frame data transfer from the host memory 60 onto the adapter 80, completion of the frame transmission itself, or other levels of transmission status.

At any instance in time, the adapter 80 may be concurrently executing some or all of the
 15 following actions: selecting the next LCP to be serviced; initiating service for LCP channel A; executing DMA fetch of data for the last frame of LCP channel B; processing a frame header and fragmentation for LCP channel C ; and, transmitting packets originated by LCP
 20 channel D.

Receive Packet Flow

Referring to Figure 8, what follows now, by way of example only, is a description of a data
 25 frame reception by an application using an RX LCP port. The operation of the ISOC 120 may vary depending on the type of protocol supported by the LCP. Handshaking between the application and the adapter 80 assumes a prior initialization performed by the LCP manager 130. The RX LCP engine 360 comprises LCP allocation logic 620, LCP Context logic 610, and DMA logic 630 all residing in the LCP domain 520. The RX processor 160, first level
 30 RX memory space 230, and RX logic 350 all straddle the boundary between the frame domain 520 and the network domain 530. The RX link logic 340 and packet assist logic 600 reside in the network domain 530. In particularly preferred embodiments of the present invention, the HeaderOut FIFO 410 is located in the first level RX memory space 230.

Frames received by the ISOC 120 from the network architecture 30 are written into LCP client buffers in the host memory 60. Availability of memory buffers is determined by the LCP RX client 100 and is indicated to the adapter 80 for insertion of incoming data frames. The LCP client 100 provides buffers by writing into a receive Doorbell on the ISOC 120, similar to the aforementioned manner in which the transmission path logic 280 is informed of new frames ready to be transmitted. The Doorbell register address is allocated such that the write operation is translated into a physical write cycle on the bus architecture 70. The adapter 80 detects the write operation and logs the new provision of empty memory areas by incrementing the number of available word entries for the specific LCP RX Client 100. The available word count is part of the related LCP context 140. Whenever an application completes processing of a received frame within a buffer, it writes to the Doorbell. The write cycle indicates the number of words in the newly available memory space. The count within the LCP context is incremented by that amount. A packet received from the network architecture 30 may be part of a larger frame that will be assembled by the adapter 80 into contiguous space in the host memory 60. Processing of received frames by the ISOC 120 generally includes the following steps:

A. Splitting packet header and data

The RX link logic 340 translates information from the network architecture 30 into a stream of packets. Each received packet is processed by the RX link logic 340 to separate the packet header from the payload data. The header is pushed into an RX HeaderIn FIFO 640 in the first level RX memory space 230. The payload is pushed into an RX data FIFO 650 in the RX logic 350. The RX data FIFO 650 may also be implemented in the first level RX memory space 230.

B. Decoding the packet header and generating and LCP frame header.

The packet header is decoded to provide fields indicative of an ID for the frame to which the packet belongs, the size of the payload, and the size of the frame data. Once the packet header is read for the RX HeaderIn FIFO 640, an indication is sent to the RX processor 160. The RX processor processes the packet header information and generates an LCP related command including information required to transfer the

packet data. Such information includes packet address and length. At the end of the header processing, a descriptor, or a set of descriptors, are written to the LCP RX HeaderOut FIFO 410, and an indication is triggered.

5 C. Transfer of data within the RX LCP Context.

The descriptors are fetched from the RX HeaderOut FIFO 410 by the RX LCP engine 360, and then decoded. The descriptors include the LCP number, packet address, packet data length and the source address of the data to be transferred in the memory subsystem 210 of the adapter 80. The RX LCP engine 340 uses the LCP Context information to create a target physical address (or addresses if a page is crossed) to be written to in the host memory 60 and initiates DMA transfers to write the data.

15 D. ISOC DMA transactions.

The ISOC 120 aims to optimize transactions on the bus architecture 70 by selecting appropriate bus commands and performing longest possible bursts.

At any instance in time, the adapter 80 may be concurrently executing some or all of the following: processing a buffer allocation for LCP channel X; initiating an inbound data write service for LCP channel A; executing a DMA store of data for LCP channel B; processing a frame assembly of a packet destined for LCP channel C; and, receiving packets for LCP channel D.

25 To minimize frame processing overhead on the RX processor 160 and TX processor 150, packet assist logic 600 comprises frame fragmentation logic, CRC and checksum calculation logic, and multicast processing logic.

In preferred embodiments of the present invention hereinbefore described, the adapter 80 is connected to the CPU 50 and memory 60 of the host computer system 10 via the bus architecture 70. However, in other embodiments of the present invention, the adapter 80 may be integrated into the host computer system 10 independently of the bus architecture 70. For

example, in other embodiment of the present invention, the adapter 80 may be integrated into the host computer system via a memory controller connected to the host memory 60.

Additionally, in preferred embodiments of the present invention hereinbefore described, the adapter 80 was implemented in the form of a pluggable adapter card for insertion into the host computer system 10. It will however be appreciated that different implementation of the adapter 80 are possible in other embodiments of the present invention. For example, the adapter 80 may be located on a mother board of the host computer system, along with the CPU 50 and the memory 60.

10

In summary, hereinbefore described by way of example of the present invention is a data communications interface for a node of a data processing network comprising a transmission channel for communicating data from the node to the network. A transmission processor connected to the transmission channel controls flow of data through the transmission channel.

15

A reception channel communicates data from the network to the node. A reception processor connected to the reception channel controls flow of data through the reception channel. A local bus provides access to a shared memory by the transmission and reception processors.